UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/844,993	04/27/2001	Jacob Dreyband	033144-017	1373

30139 7590 02/01/2008
WILSON & HAM
2530 BERRYESSA ROAD
PMB: 348
SAN JOSE, CA 95132

EXAMINER	
PHAM, KHANH B	

ART UNIT	PAPER NUMBER
2166	

MAIL DATE	DELIVERY MODE
02/01/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte JACOB DREYBAND, LEONID NILVA, and
MICHAEL SHAPIRO

Appeal 2007-3537
Application 09/844,993
Technology Center 2100

Decided: February 1, 2008

Before JEAN R. HOMERE, ST. JOHN COURTENAY III, and
STEPHEN C. SIU, *Administrative Patent Judges*.

SIU, *Administrative Patent Judge*.

DECISION ON APPEAL

I. STATEMENT OF THE CASE

Appellants appeal under 35 U.S.C. § 134(a) from the Examiner's Final Rejection of claims 1-36. We have jurisdiction under 35 U.S.C. § 6(b). We affirm.

A. Invention

1The invention at issue relates to mapping a descriptive language to an object oriented data presentation (Spec. 3). In particular, a Schema or data description that describes relational structural complexities is mapped into an executable object oriented programmable representation using internal static classes (*id.* 8). The object oriented data presentation includes the internal static class that corresponds to the structure complexities of the data description (*id.*).

B. Illustrative Claims

Claim 1, which further illustrates the invention, follows.

> 1. A method for mapping a descriptive language including a data description having a structure complexity into an object oriented programming language, comprising:
>
> receiving the data description;
>
> identifying a complex-type element in the data description; and
>
> creating an executable object oriented class that is independently executable in any of a plurality of run-time environments and that corresponds to the identified complex-type element, wherein the class includes an internal static class, wherein the internal static class corresponds to the structure complexity of the data description.

C. Rejection

Claims 1-3, 8, 9, 14-17, 22, 23, 28, 29, and 36 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Grady Booch, et al., UML for XML Schema Mapping Specification, December 8, 1999 ("Grady") and U.S. Patent Publication No. 2002/0133811 A1 ("Duftler").[1] Claims 4-7, 10-

[1] Although addressed in a separate paragraph in the Examiner's Answer

13, 18-21, 24-27, and 32-35 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Grady, Duftler, and U.S. Patent No. 6,083,276 ("Davidson").[2]

PRINCIPLES OF LAW

"What matters is the objective reach of the claim. If the claim extends to what is obvious, it is invalid under *§ 103*." *KSR Int'l Co. v. Teleflex, Inc.*, 127 S. Ct. 1727, 1742 (2007). To be nonobvious, an improvement must be "more than the predictable use of prior art elements according to their established functions." *Id.* at 1740. Appellant has the burden on appeal to the Board to demonstrate error in the Examiner's position. *See In re Kahn*, 441 F.3d 977, 985-86 (Fed. Cir. 2006) ("On appeal to the Board, an applicant can overcome a rejection [under § 103] by showing insufficient evidence of *prima facie* obviousness or by rebutting the *prima facie* case with evidence of secondary indicia of nonobviousness.") (quoting *In re Rouffet*, 149 F.3d 1350, 1355 (Fed. Cir. 1998)). Therefore, we look to Appellant's Brief to show error in the proffered prima facie case.

II. CLAIM GROUPING

(Ans. 8), claims 8, 14, 22, 28, and 36 are subject to the same ground of rejection as claim 1.

[2] The Examiner finds claims 4-14, 18-28, and 32-36 unpatentable over Grady, Duftler, and Davidson (Ans. 12) but presents arguments for unpatentability of only claims 4-7, 10-13, 18-21, 24-27, and 32-35 over Grady, Duftler, and Davidson while presenting arguments for unpatentability of claims 8, 9, 14, 22, 23, 28, and 36 over Grady and Duftler (Ans. 3-12). We therefore consider the issue of unpatentability of claims 8, 9, 14, 22, 23, 28, and 36 over Grady and Duftler.

1"When multiple claims subject to the same ground of rejection are argued as a group by appellant, the Board may select a single claim from the group of claims that are argued together to decide the appeal with respect to the group of claims as to the ground of rejection on the basis of the selected claim alone. Notwithstanding any other provision of this paragraph, the failure of appellant to separately argue claims which appellant has grouped together shall constitute a waiver of any argument that the Board must consider the patentability of any grouped claim separately." 37 C.F.R. § 41.37(c)(1)(vii) (2005).[3]

Here, Appellants argue claims 1-3, 8, 9, 14-17, 22, 23, 28-31, and 36, which are subject to the same ground of rejection, as a first group (App. Br. 5-6) and claims 4-7, 10-13, 18-21, 24-27, and 32-35 as a second group.

III. CLAIMS 1-3, 8, 9, 14-17, 22, 23, 28-31, AND 36

We select claim 1 as the sole claim on which to decide the appeal of the first group. Rather than reiterate the positions of parties *in toto*, we focus on the issue(s) therebetween.

Appellants dispute the Examiner's finding of obviousness and argue that the "(Java) code of Duftler is still not *independently* executable in a run-time environment" (App. Br. 10) because "language in Duftler is only executable in a Java run-time environment" (*id.* 10-11).

In response, the Examiner asserts that Duftler discloses "Java specifically defining and implementing XML" and "implementing JavaBeans components using any scripting language or languages that corresponds to any of a plurality of object oriented languages executable in

[3] We cite to the version of the Code of Federal Regulations in effect at the time of the Appeal Brief. The current version includes the same rules.

run-time environments" (Ans. 8). The Examiner further equates Java and the implementation of JavaBeans components with an "'object oriented class that is independently executable in any of a plurality of run-time environments' [page 1, col 1-2, 0009-0014]" (*id.*) as recited in claim 1.

Based on the record before us, we agree with the Examiner's finding that the Java code of Duftler is independently executable in any of a plurality of run-time environments. Duftler discloses XML-based (Bean Scripting Components BSC) documents compiled into a Java class which is "100% compatible with any other JavaBeans component implemented directly in Java itself (¶ [0014]). Thus, Duftler discloses that BSC defines JavaBeans components implemented in Java. One of ordinary skill in the art would have readily recognized that Java is independently executable in a plurality of run-time environments, as Java is compiled to platform-independent byte code that is capable of being interpreted and executed by a Java Virtual Machine on multiple platforms. Appellants have failed to establish otherwise.

Appellants further argue that "there is no motivation for one skilled in the art . . . to combine the teachings in Duftler (with that of Grady) since Duftler merely teaches employing XML (a non-Schema language) in a specific descriptive function" (App. Br. 11-12). Based on the record before us, we find that Grady discloses XML Schema mapping where the XML Schema defines "the structure of XML document instances that belong to a specific document type" (Abstract, page 1). In addition, Grady discloses that XML is used to generate "code associated with implementing" processes (Introduction, page 1). Therefore, Grady discloses a design language (XML) and mapping XML Schemas to "rapidly generate" code for implementing processes. In other words, Grady demonstrates that using a design language mapping to generate code for process implementation was known in the art.

Duftler discloses an interface design language (i.e., BSC) for implementing JavaBeans (¶ [0013]). Therefore, Duftler demonstrates that implementing processes with an interface design language was known in the art.

We note that in *KSR*, the Supreme Court reaffirmed that "[w]hen a patent 'simply arranges old elements with each performing the same function it had been known to perform' and yields no more than one would expect from such an arrangement, the combination is obvious." *KSR*, 127 S. Ct. at 1740 (quoting *Sakraida v. Ag Pro, Inc.*, 425 U.S. 273, 282 (1976)). In the present case, there would have been a finite number of ways in which one of ordinary skill in the art would have implemented processes using information from corresponding design languages. Using an XML schema as disclosed by Grady to implement processes in a plurality of Java run-time environments of Duftler would have been known to one of ordinary skill in the art by arranging elements of Grady and Duftler to perform the same function of implementing the associated processes. Such a rearrangement of design languages would have yielded no more that expectedly predictable results of process implementation. "When there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp. If this leads to the anticipated success, it is likely the product not of innovation but of ordinary skill and common sense." *KSR Int'l Co. v. Teleflex Inc.*, 127 S. Ct. 1727, 1742 (2007).

Appellants also argue that "(t)here is nothing in Duftler that suggests abandoning the XML descriptive language technique taught therein to try to create JavaBeans with a Schema language, and to do so would likely render the intended purpose of Duftler useless" (App. Br. 12).

Based on the record before us, we find that although Appellants assert that Duftler's intended purpose would have been rendered "useless" by combination with Grady, Appellants have nevertheless failed to demonstrate that Duftler's intended purpose of implementing processes by using corresponding design languages would have been rendered useless by utilizing the XML Schemas of Grady. Indeed, Grady discloses process implementation using XML Schemas as set forth above.

It follows that Appellants have failed to demonstrate that the Examiner erred in rejecting claim 1. Therefore, we affirm the rejection of claim 1, and of claims 2, 3, 8, 9, 14-17, 22, 23, 28-31, and 36, which fall therewith.

IV. CLAIMS 4-7, 10-13, 18-21, 24-27, AND 32-35

We select claim 4 as the sole claim on which to decide the appeal of the second group. Rather than reiterate the positions of parties *in toto*, we focus on the issue(s) therebetween.

Appellants argue that "the pending independent claims are not obvious over the combination of Grady and Duftler" and that "Davidson does nothing to cure these deficiencies of Grady and Duftler" (App. Br. 12). As set forth above, we agree with the Examiner that independent claim 1, from which claim 4 depends, is unpatentable over the combination of Grady and Duftler.

It follows that Appellants have failed to demonstrate that the Examiner erred in rejecting claim 4. Therefore, we affirm the rejection of claim 4, and of claims 3-7, 10-13, 18-21, 24-27, and 32-35, which fall therewith.

V. ORDER

In summary, we affirm the rejections of claims 1-36 under § 103(a).

No time for taking any action connected with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED

ce

WILSON & HAM
2530 BERRYESSA ROAD
PMB: 348
SAN JOSE, CA 95132